MASTER'S THESIS

Master of Science in Advanced Mathematics and Mathematical Engineering

Title: Numerical Solution of the Helmholtz Equation using High-Order Continuous Galerkin Methods

Author: Sandra Corella Pérez

Advisor: Josep Sarrate Ramos and Eloi Ruiz Gironés

Department: Departament d'Enginyeria Civil i Ambiental (751)

Academic year: 2016-2017

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering

Master's thesis

Numerical Solution of the Helmholtz Equation using High-Order Continuous Galerkin Methods

Sandra Corella Pérez

Supervised by Josep Sarrate Ramos and Eloi Ruiz Gironés

June, 2017

Abstract

We show a continuous Galerkin formulation with high-order polynomials to solve the Helmholtz equation. High-order formulations obtain solutions with less numerical error, and can use curved high-order meshes to approximate the domain. To reduce the computational requirements of the high-order formulation, we apply a static condensation technique. Using this technique, we eliminate a set of unknowns from the global linear system and therefore, we solve a smaller system of equations. Then, we recover the full solution by solving several systems that only involve the unknowns of a single element. In the examples we show that the proposed implementation converges optimally to the analytical solution both for two and three dimensional examples. We also show an application where the mesh is curved in order to better capture the geometry.

Keywords

Numerical modeling, finite element method, Helmholtz equation, high-order, static condensation

1. Introduction

Physical problems related to waves, for instance accoustic problems [5], are usually modelled using Helmholtz equation. Helmholtz equation is a second order partial differential equation that represents a time-indepent form of the wave equation. Helmholtz equation has been solved analitically for many basic shapes but no closed form solutions are known for complex shapes. Therefore, numerical methods should be performed in order to find approximated solutions to the Helmholtz equation.

Some authors, like [5], study the finite element method (FEM) solution of the Helmholtz equation. Finite element method [3, 4, 12], in particular continuous Galerkin method, provides a continuous piecewise polynomial approximation of the solution which minimizes the residual of the solution [9].

Continuous Galerkin formulations with high-order polynomials have exponential convergence of the numerical error and can be used with curved high-order meshes. However, they involve larger linear systems than low order formulations, with more dense matrices.

Thus, to improve the efficiency of high-order formulations it is necessary to apply a static condensation (hybridization) technique [9]. The idea consists in expressing the element unknowns in terms of an hybrid unknown which is defined only in the boundaries of the elements. This gives a global system of equations that only involves hybrid unknowns. Then, by solving a series of small linear systems involving only the unknowns of a single element, the full solution is recovered.

The objective of this master thesis is to solve the Helmholtz equation numerically using high-order finite element methods in 2D and 3D. Standard continuous Galerkin (CG) finite element method will be considered as well as, its hybridized version, hybridizable continuous Galerkin method (HCG). The accuracy and performance of the implemented numerical methods will be verified with some examples and a numerical convergence analysis will be performed to verify the order of convergence and therefore, validate the proposed implementation.

Section 2 derives the strong form of the Helmholtz equation from the wave equation formulation. After that the weak form is found and discretized. Section 3 introduces the linear system of equations to be solved, numerical integration, static condensation technique and some implementation details. Section 4 presents two examples and a numerical convergence analysis. Later in this section, a real case example is performed. Finally at section 5 conlcusions and future work are presented.

2. Finite Element Method Formulation

Wave problems are usually modeled using the wave equation

$$\nabla^2 U(\mathbf{x}, t) - \frac{1}{c^2}\frac{\partial^2 U(\mathbf{x}, t)}{\partial t^2} = F(\mathbf{x}, t), \tag{1}$$

where ∇^2 is the laplacian, $U(\mathbf{x}, t)$ is the solution, the source term is $F(\mathbf{x}, t)$ and c is a constant that in some problems is related to the propagation velocity.

The Helmholtz equation is a partial differential equation which represents a time-independent form of the wave equation (1). It can be obtained from the wave equation (1) by expressing the solution $U(\mathbf{x}, t)$ and the source term $F(\mathbf{x}, t)$ as the product of two separated terms: one time-dependent and another one time-independent. Specifically

$$U(\mathbf{x}, t) = u(\mathbf{x})e^{-i\omega t}, \tag{2}$$

$$F(\mathbf{x}, t) = f(\mathbf{x})e^{-i\omega t}. \tag{3}$$

Substituting equations (2)-(3) into the wave equation (1) the Helmholtz equation is obtained as a time-independent partial differential equation

$$\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = f(\mathbf{x}), \tag{4}$$

where k represents the wave number and is computed as $k = \omega/t$.

2.1 Strong Form

Let $\Omega \subset \mathbb{R}^n$ be an open bounded domain with boundary $\partial\Omega$, defined as $\partial\Omega = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ where Γ_D, Γ_N and Γ_R are disjoint. The strong form for the second-order elliptic Helmholtz equation with Dirichlet, Neumann and Robin boundary conditions are written as follows:

$$\begin{cases} \nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = g_d(\mathbf{x}) & \mathbf{x} \in \Gamma_D, \\ \nabla u(\mathbf{x}) \cdot \mathbf{n} = g_N(\mathbf{x}) & \mathbf{x} \in \Gamma_N, \\ \nabla u(\mathbf{x}) \cdot \mathbf{n} - i\tau k u(\mathbf{x}) = g_R(\mathbf{x}) & \mathbf{x} \in \Gamma_R, \end{cases} \tag{5}$$

where $\mathbf{n}$ is the outward unit normal vector to the boundary $\partial\Omega$, k is the wave number and τ is the transmission coefficient. Note that Neumann boundary conditions can be represented by Robin conditions setting the transmission coefficient $\tau = 0$.

2.2 Weak Form

We define the following function spaces:

$$H^1(\Omega, \mathbb{C}) = \left\{ u \;\middle|\; \int_\Omega u\tilde{u}\, \mathrm{d}\Omega < \infty, \int_\Omega \nabla u \nabla \tilde{u}\, \mathrm{d}\Omega < \infty \right\} \tag{6}$$

$$V_0 = \left\{ v \in H^1(\Omega, \mathbb{C}) \;\middle|\; v_{|\Gamma_D} = 0 \right\}, \tag{7}$$

where $\tilde{\cdot}$ is the complex conjugate, and the set of admissible functions:

$$V_D = \left\{ u \in H^1(\Omega, \mathbb{C}) \;\middle|\; u_{|\Gamma_D} = g_d \right\}. \tag{8}$$

Given the test function $v \in V_0$ which is zero on Dirichlet boundary, we multiply equation (5) by v and integrate over the domain

$$\int_\Omega \nabla^2 u \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega. \tag{9}$$

We substitute (10) into (9) and we get:

$$\nabla \cdot (\nabla u \tilde{v}) = \tilde{v} \nabla \cdot (\nabla u) + \nabla u \nabla \tilde{v} \tag{10}$$

$$\int_\Omega \nabla(\nabla u \tilde{v}) \, \mathrm{d}\Omega - \int_\Omega \nabla u \nabla \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega. \tag{11}$$

Applying the divergence theorem on the first integral of equation (11) we get

$$\int_\Gamma \nabla u \mathbf{n} \tilde{v} \, \mathrm{d}\Omega - \int_\Omega \nabla u \nabla \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega. \tag{12}$$

The boundary is decomposed as $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_R$ thus,

$$\int_{\Gamma_D} \nabla u \cdot \mathbf{n} \tilde{v} \, \mathrm{d}\Omega + \int_{\Gamma_N} \nabla u \cdot \mathbf{n} \tilde{v} \, \mathrm{d}\Omega + \int_{\Gamma_R} \nabla u \cdot \mathbf{n} \tilde{v} \, \mathrm{d}\Omega - \int_\Omega \nabla u \nabla \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega. \tag{13}$$

Since v vanishes on the Dirichlet boundary and applying Neumann and Robin boundary conditions we have:

$$\int_{\Gamma_N} g_N \tilde{v} \, \mathrm{d}\Gamma + i \int_{\Gamma_R} \tau k u \tilde{v} \, \mathrm{d}\Gamma + \int_{\Gamma_R} g_R \tilde{v} \, \mathrm{d}\Gamma - \int_\Omega \nabla u \nabla \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega. \tag{14}$$

Sorting terms and grouping the unknowns on the left hand side we get that the weak form equivalent to (5) is:

Find $u \in V_D$ such that

$$-\int_\Omega \nabla u \nabla \tilde{v} \, \mathrm{d}\Omega + k^2 \int_\Omega u \tilde{v} \, \mathrm{d}\Omega + i\tau k \int_{\Gamma_R} u \tilde{v} \, \mathrm{d}\Gamma = \int_\Omega f \tilde{v} \, \mathrm{d}\Omega - \int_{\Gamma_N} g_N \tilde{v} \, \mathrm{d}\Gamma - \int_{\Gamma_R} g_R \tilde{v} \, \mathrm{d}\Gamma, \tag{15}$$

for all $v \in V_0$

2.3 Discretization

Let's assume that Ω is discretized in n_e disjoint elements e_i with boundary ∂e_i.

$$\Omega = \bigcup_{i=1}^{n_e} e_i, \quad e_i \cap e_j = \emptyset \text{ for } i \neq j.$$

For each physical element, e, we use a reference element, e^R, in order to define the shape functions. The reference quadrilateral element is $[-1, 1]^2$, and the reference hexahedral element is $[-1, 1]^3$.

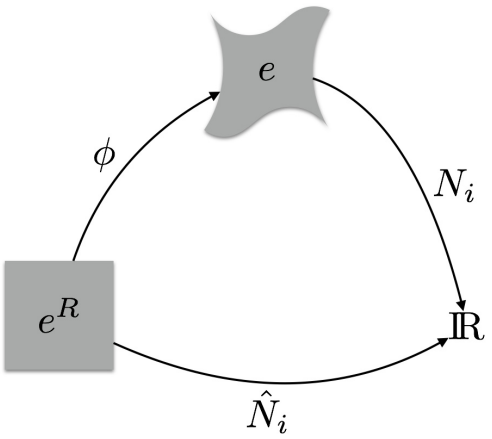


Figure 1: Iso-parametric mapping

The corresponding discretized spaces equivalent to H^1 and V_0 from equations (6)-(8) are:

$$V^h = \left\{ u \in C^0(\Omega, \mathbb{C}) \middle| u \circ \phi_{|e^R} \in P^p(e^R) \right\}, \tag{16}$$

$$V_0^h = \left\{ v \in V^h \middle| v_{|\Gamma_d} = 0 \right\}, \tag{17}$$

where $P^p(e)$ is the space of polynomials of degree at most p in the reference element e^R.

Let $\hat{N}_i$, $i = 1, \dots, n_n$, be a Lagrangian basis of a continouos element-wise polynomial of the space V^h (16), where n_n is the number of nodes. That is, the N_i are the standard shape functions of continuous Galerkin formulations defined on the nodes of the element. We assume that the numerical approximation to the analytical solution belongs to V^h.

The physical element is written is terms of the reference element using $\phi = \sum_i^{n_n} \mathbf{x}_i \hat{N}_i$. Being ϕ^e the usual iso-parametric mapping defined in finite element formulation. Figure 1 shows the reference element mapped into the physical one by means of the iso-parametric mapping.

The shape functions in the physical element N_i are written in terms of the iso-parametric mapping as $N_i = \hat{N}_i \circ \phi^{-1}$. Therefore, the gradient of the shape functions is written as $\nabla \hat{N}_i = (\nabla N_i) \circ \phi D\phi \implies \nabla \hat{N}_i (D\phi)^{-1} = (\nabla N_i) \circ \phi$.

Then, the discretized solution $u \in V^h$ is written as

$$u = \sum_{j=1}^{n_u} N_j u_j + \sum_{k=n_u+1}^{n_n} N_k u_k,$$

where n_n is the number of nodes and n_u is the number of nodes where the solution is unknown. Without loss of generality, we assume that the unknowns related to the Dirichlet nodes are the first n_u unknowns. Thus, the discrete weak formulation equivalent to (15) is to find $u_j \in \mathbb{C}$ for $j = 1, \dots, n_n$ such that

$$\sum_{j=1}^{n_u}\left[-\int_\Omega \nabla N_i \nabla N_j \,\mathrm{d}\Omega + k^2\int_\Omega N_i N_j \,\mathrm{d}\Omega + i\tau k\int_{\Gamma_R} N_i N_j \,\mathrm{d}\Gamma\right] u_j \;=$$

$$\int_\Omega f N_i \,\mathrm{d}\Omega - \int_{\Gamma_N} g_N N_i \,\mathrm{d}\Gamma - \int_{\Gamma_R} g_R N_i \,\mathrm{d}\Gamma \;+$$

$$\sum_{k=n_u+1}^{n_n}\left[\int_\Omega \nabla N_i \nabla N_k \,\mathrm{d}\Omega - k^2\int_{\Gamma_N} N_i N_k \,\mathrm{d}\Gamma + i\tau k\int_{\Gamma_R} N_i N_k \,\mathrm{d}\Gamma\right] u_k \tag{18}$$

for all N_i, with $i = 1, \ldots, n_u$.

3. Finite Element Method Implementation

3.1 Linear System

Equation (18) is a linear system of equations that can be written in a matrix form as

$$\mathbf{Au} = \mathbf{b}, \tag{19}$$

where

$$\mathbf{A} = \mathbf{K} + \mathbf{M} + \mathbf{M}_c, \tag{20}$$

$$\mathbf{b} = \mathbf{f} + \mathbf{f_D} + \mathbf{f_N} + \mathbf{f_R}. \tag{21}$$

Each one of this matrices and vectors are computed in terms of the shape functions, its gradients and the prescribed boundary conditions as follows:

$$\begin{aligned}
\mathbf{K} &= \int_\Omega \nabla N_i \nabla N_j \, \mathrm{d}\Omega, \\
\mathbf{M} &= k^2 \int_\Omega N_i N_j \, \mathrm{d}\Omega, \\
\mathbf{M}_c &= i\tau k \int_{\Gamma_R} N_i N_j \, \mathrm{d}\Omega, \\
\mathbf{f} &= \int_\Omega f N_i \, \mathrm{d}\Omega, \\
\mathbf{f}_N &= -\int_{\Gamma_N} g_N N_i \, \mathrm{d}\Omega, \\
\mathbf{f}_R &= -\int_{\Gamma_R} g_R N_i \, \mathrm{d}\Omega, \\
\mathbf{f}_D &= \int_\Omega \nabla N_i \nabla N_j \, \mathrm{d}\Omega + k^2 \int_\Omega N_i N_k \, \mathrm{d}\Gamma + i\tau k \int_\Omega N_i N_k \, \mathrm{d}\Gamma.
\end{aligned}$$

3.2 Numerical Integration

The integrals over the domain Ω will be computed as the sum of the integrals over the elements. For instance, let f in $H^1(\Omega, \mathbb{C})$ be a function, then the integral over tge domain Ω is

$$\int_\Omega f \, \mathrm{d}\Omega = \sum_{e \in \Omega} \int_e f \, \mathrm{d}\Omega.$$

All the element integrals will be computed at the reference element e^R. The integral of f at the physical element will be computed at the reference element using the iso-parametric mapping ϕ as $f : \mathbb{R} \to \mathbb{C}$ defined in finite element formulation:

$$\int_e f \, \mathrm{d}\Omega = \int_{e^R} f \circ \phi |D\phi| \, \mathrm{d}\xi.$$

For instance, the mass matrix $\mathbf{M}$ is integrated as follows:

$$M_{ij} = k^2 \int_e N_i N_j \, \mathrm{d}\Omega = k^2 \int_{e^R} \big(N_i \circ \phi\big)\big(N_j \circ \phi\big) |D\phi| \, \mathrm{d}\xi = k^2 \int_{\Omega^M} \hat{N}_i \hat{N}_j |D\phi| \, \mathrm{d}\xi,$$

and, the ij-component of stiffness matrix $\mathbf{K}$ is:

$$K_{ij} = \int_e \nabla N_i \nabla N_j \, \mathrm{d}\Omega = \int_{e^R} \big(\nabla N_i \circ \phi\big)\big(\nabla N_j \circ \phi\big) |D\phi| \, \mathrm{d}\xi = \int_{\Omega^M} \nabla \hat{N}_i \nabla \hat{N}_j |D\phi| \, \mathrm{d}\xi.$$

Once the integrals are formulated on the reference element, we approximate them using a Gauss quadrature as follows:

$$\int_{e^R} g(\xi) \, \mathrm{d}\xi \approx \sum_{k=1}^{n_g} g(\xi_{\mathbf{k}}) w_k,$$

where n_g is the number of integration points and w_k, for $k = 1, \cdots, n_g$, is the weight associated to each integration point ξ_k.

3.3 Static Condensation

In order to reduce the size of the linear system to be solved, we apply the static condensation or hybridization technique [9]. Without lose of generality, we assume that the unknowns are ordered in such a way that $\mathbf{u}^T = ((\mathbf{u}^b)^T, (\mathbf{u}^i)^T)$, where $\mathbf{u}^b$ are the unknowns associated to the nodes at the element boundaries, and $\mathbf{u}^i$ are the unknowns associated to the inner nodes of the elements. Figure 2 shows a quadrilateral mesh with 4 elements and degree $p = 2$. The blue points are the boundary element nodes, $\mathbf{u}^b$, and orange diamonds represent the inner element nodes, $\mathbf{u}^i$.

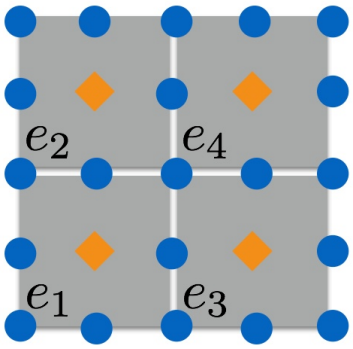


Figure 2: Mesh in $\mathbb{R}^2$ with 4 elements, degree $p = 2$

Reordering with the new numeration, the linear system of equations is written as follows

$$\begin{bmatrix} \mathbf{A}^{bb} & \mathbf{A}^{bi} \\ \mathbf{A}^{ib} & \mathbf{A}^{ii} \end{bmatrix} \begin{pmatrix} \mathbf{u}^b \\ \mathbf{u}^i \end{pmatrix} = \begin{pmatrix} \mathbf{f}^b \\ \mathbf{f}^i \end{pmatrix}. \tag{22}$$

From the second equation of the system 22 we can write $\mathbf{u}^i$ in terms of $\mathbf{u}^b$

$$\mathbf{A}^{ib}\mathbf{u}^b + \mathbf{A}^{ii}\mathbf{u}^i = \mathbf{f}^i, \tag{23}$$

which implies that

$$\mathbf{u}^i = (\mathbf{A}^{ii})^{-1}[\mathbf{f}^i - \mathbf{A}^{ib}\mathbf{u}^b]. \tag{24}$$

Substituting the equation (24) into the first equation of (22) we can express the solution in terms of the boundary nodes

$$\mathbf{A}^{bb}\mathbf{u}^b + \mathbf{A}^{bi}(\mathbf{A}^{ii})^{-1}[\mathbf{f}^i - \mathbf{A}^{ib}\mathbf{u}^b] = \mathbf{f}^b. \tag{25}$$

Reorganizing terms, we obtain a linear system of the form:

$$\left[\mathbf{A}^{bb} - \mathbf{A}^{bi}(\mathbf{A}^{ii})^{-1}\mathbf{A}^{ib}\right]\mathbf{u}^b = \mathbf{f}^b - \mathbf{A}^{bi}(\mathbf{A}^{ii})^{-1}\mathbf{f}^i \tag{26}$$

Notice that the system associated involves only boundary nodes which essentially means that, according to equation (24), the unknowns belonging to the interior of an element are put in terms of the unknowns on the boundary of the same element.

Once the boundary solution is found for each element using equation (26), the unknowns related to the inner nodes are recovered by means of

$$\mathbf{u}^i = (\mathbf{A}^{ii})^{-1}\mathbf{f}^i - (\mathbf{A}^{ii})^{-1}\mathbf{A}^{ib}\mathbf{u}^b. \tag{27}$$

Notice that the inverse form $(\mathbf{A}^{ii})^{-1}$ has to be computed. However, $\mathbf{A}^{ii}$ is a block diagonal matrix and each block only involves the inner nodes of a single element. Then, $\mathbf{A}^{ii}$ can be easily inverted element by element. Since A^{ii} is block diagonal, we construct the condensed linear system

$$A_C\mathbf{u}_e^b = b_C \tag{28}$$

by assembling the condensed local contributions. Where

$$A_C = \left[\mathbf{A}_e^{bb} - \mathbf{A}_e^{bi}(\mathbf{A}_e^{ii})^{-1}\mathbf{A}_e^{ib}\right], \tag{29}$$

$$b_C = \mathbf{f}_e^b - \mathbf{A}_e^{bi}(\mathbf{A}_e^{ii})^{-1}\mathbf{f}_e^i. \tag{30}$$

Note that to compute the elemental contributions, we need to invert each block A_e^{ii}.

The number of unknowns of the system (26) is reduced respect to the complete system. The total number of unknowns is the sum of boundary and inner nodes $n_u = n_i + n_b$. Using the standard procedure, we solve a linear system involving all the n_u unknowns, whereas using static condensation, we only solve for n_b unknowns.

We are considering high-order polynomials so when the degree is increased the number of inner nodes increase with order p^d while the number of boundary nodes increase in order p^{d-1}. In the particular case of $\mathbb{R}^2$ the number of boundary nodes for a single element is $2(p+1) + 2(p-1) = 4p$ and the number of inner nodes is $(p-1)\cdot(p-1) \approx p^2$.

Due to that fact, the new condensed system presents an advantadge. With the static condensation procedure we will solve a smaller linear system involving only the boundary unknowns. Then, for each element we will solve a small independent linear system to compute u_e^i. Since the system is obtained by assembling elemental contributions, it is not necessary to build the global system involving the boundary and inner nodes and therefore, the amount of memory usage is also reduced. To take advantadge of the computational cost of calculate the form $(\mathbf{A}_e^{bi}\mathbf{A}_e^{ii})^{-1}$, the product is computed once and saved to be used to solve the linear system of each element.

3.4 Implementation Details and Execution Details

The code to solve the Helmholtz equation by means of finite element method has been implemented using Python language[10] and taking advantage of Numpy [11] package to perform operations of multi-index arrays, and Scipy [8] package to solve the linear systems of equations. Anaconda distribution [1] of Python has been used because Numpy and Scipy packages are already compiled against the Math Kernel Library (MKL)[6].

Math Kernel Library implements routines from the LAPACK package [7] that are used for solving systems of linear equations, linear least squares problems, eigenvalue and singular value problems, performing a number of related computational tasks and enables High Performance computing. The MKL library includes LAPACK routines for both real and complex data.

The linear systems of equations (19) and (28) are solved using the LU decomposition method implemented by the SuperLU library [2]. To this end, the Scipy package provides an interface to call the SuperLU library from the python code.

All the executions of the code have been done in a ubuntu server with six core Intel-Xeon with 20Gb of memory and SSD hard disk.

The elapsed time measured in all the examples is the cpu time and it is measured in seconds.

4. Numerical Examples

In this section we present several examples to illustrate the performance and accurancy of the finite element methods, with and without static condensation, presented in the previous sections to solve the Helmholtz equation. The objective of the Example 1 is to validate the implementation when $\mathbb{R}^2$ meshes are considered and Dirichlet boundary conditions are prescribed. Example 2 validates the implementation when $\mathbb{R}^3$ meshes are considered and Robin boundary conditions are prescribed. Later convergence analysis validates that the implementation performs well and the the order of convergence is what we expect. Once the implementation is validated, example 3 shows the solution of the Helmholtz equation for a curved domain discretized with high-order quadrilaterals.

To measure the accurancy of the solutions we define the L^2 norm as

$$E^2 = ||u - u_h||^2_{L^2} = \int_\Omega (u - u_h)(\widetilde{u - u_h})\,\mathrm{d}\Omega = \sum_e \int_{\Omega^e} (u - u_h)(\widetilde{u - u_h})\,\mathrm{d}\Omega \tag{31}$$

where u is the analytical solution and u_h is the numerical solution. In that case L^2 norm can be computed in examples 1 and 2 since the solution $u(\mathbf{x})$ is known.

4.1 Example 1

For the first example we consider the follwing problem

$$\begin{cases} \nabla^2 u + k^2 u = f & \text{in } \Omega \\ u = g_d & \text{on } \Gamma_D \end{cases}$$

where $\Omega = [0,1] \times [0,1]$ and only Dirichlet conditions are considered. The source term is $f(\mathbf{x}) = 0$, the wave number is $k = 24\pi$ and two incident wave directions considered which are $\mathbf{v}_1 = [1,0]$ and $\mathbf{v}_2 = [0.866, 0.5]$. The Dirichlet condition is set in order to obtain an analytical solution of the form

$$u(\mathbf{x}) = e^{ik\mathbf{v}\cdot\mathbf{x}}.$$

The domain is discretized using 1024 elements and the size of each element is $h = 1/2^5 = 0.03125$. The numerical solution using continuous Galerkin formulation with and without static condensation have been computed using polynomials of degrees $p = 1, \ldots, 5$. Table 1 shows the L_2-norm and elapsed time of CG and HCG methods in the case that the direction of the incident wave is $\mathbf{v}_1 = [1,0]$. Figures 3 and 4 show the solution por the polynomial degrees $p = 1$, $p = 3$ and $p = 5$ using CG and HCG methods, respectively. The same results are shown in the case that the direction of the indent wave is $\mathbf{v}_1 = [0.866, 0.5]$ at table 2 and figures 6 and 5.

Results presented show that the HCG metod reproduces the same results that CG method. This is because the same linear system of equations is solved using CG and HCG method. Therefore, the solution is the same up to round-off errors. Tables 1 and 2 show that CG and HCG errors are the same as we expected.Figures 3-4, for the horizontal incident wave, and 6-5, in the case of the diagonal incident wave, also show that the solution is the same independently of using CG or HCG method. Since the solutions are

			CG		HCG	
Degree	Nodes	Elements	Error	Time (s)	Error	Time (s)
1	1089	1024	1.397E+00	3.754E-01	1.397E+00	4.81E-01
2	4225	1024	7.321E-01	1.32E+00	7.321E-01	9.829E-01
3	9409	1024	2.852E-01	6.089E+00	2.852E-01	3.247E+00
4	16641	1024	1.167E-02	1.457E+01	1.167E-02	8.279E+00
5	25921	1024	2.019E-04	2.782E+01	2.019E-04	1.485E+01

Table 1: CG and HCG solution with $p = 1, \dots, 5$, $h = 1/2^5$ and $\mathbf{v}_1 = [1, 0]$

			CG		HCG	
Degree	Nodes	Elements	Error	Time (s)	Error	Time (s)
1	1089	1024	4.636E+00	3.441E-01	4.636E+00	4.858E-01
2	4225	1024	4.521E+00	1.034E+00	4.521E+00	8.411E-01
3	9409	1024	1.991E-02	6.611E+00	1.991E-02	2.741E+00
4	16641	1024	6.441E-04	1.214E+01	6.441E-04	6.269E+00
5	25921	1024	4.605E-05	2.233E+01	4.605E-05	1.512E+01

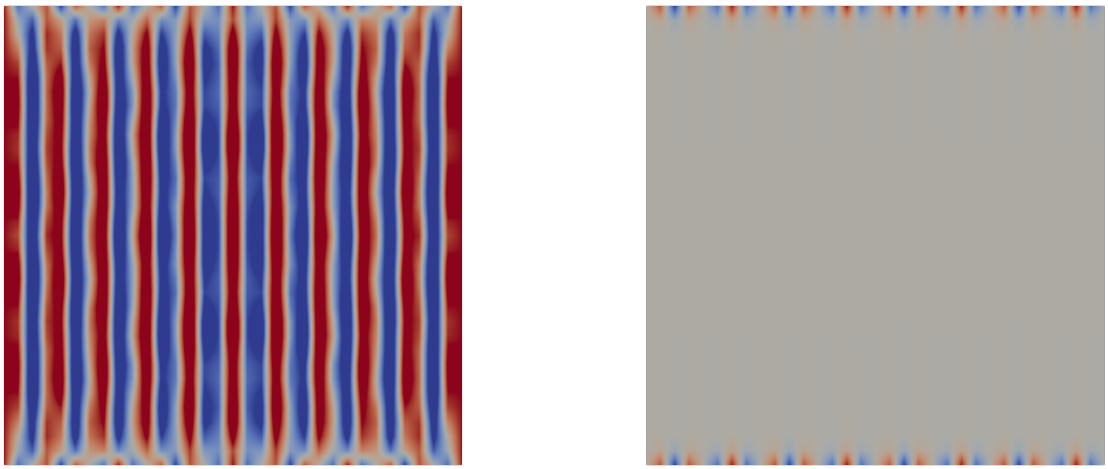
Table 2: CG and HCG solution with $p = 1, \dots, 5$, $h = 1/2^5$ and $\mathbf{v}_2 = [0.866, 0.5]$

the same using CG and HCG only CG figures will be comented.

Figures 3a and 3b show that the mesh with interpolation degree $p = 1$ has not enough resolution to capture the solution which has frequency $k = 24\pi$. When interpolation degree is $p = 3$, the real part of the solution has small error but the complex part is not well represented as we can see at figures 3c and 3d. When the interpolation degree is $p = 5$, the solution is captured with small errors as can be seen at figures 3e and 3f.

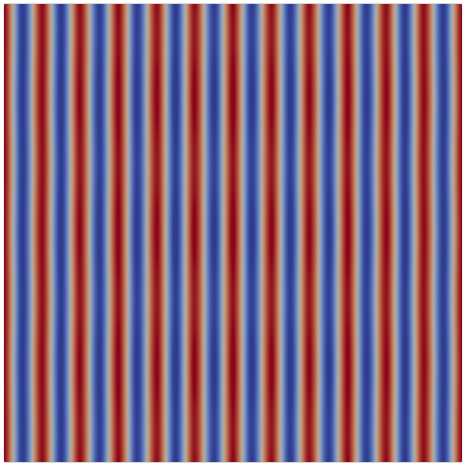
Note that, when the incident wave has direction $\mathbf{v}_2 = [0.866, 0.5]$ the interpolation degree $p = 3$, see figure 6d, gives us better complex solution than when the incident wave has direction $\mathbf{v}_1 = [1.0, 0.0]$, see figure 3d. When the interpolation degree is $p = 5$, the solution is captured with small errors as can be seen at figures 6e and 6f. When $p = 5$ there isn't significant differences between the results changing the incident wave direction.

Results show that the error decreases as the degree increases and the element size decresases. To be able to find a solution with small error when the frequency is $k = 24\pi$ we need at least degree 4. The error is lower when the incident wave direction is $\mathbf{v}_2 = [0.866, 0.5]$ instead of $\mathbf{v}_1 = [1, 0]$. This could be because when the incident wave direction is $\mathbf{v}_2 = [0.866, 0.5]$ the solution is better aligned with the nodes of the mesh. The results also show that as the polynomial degree increases, the elapsed time of CG method is greater than HCG method. This is because HCG method solves a smaller linear system of equations. Moreover, the memory usage in HCG is lower than in the case of CG method. For $p = 1$ the ellapsed time of HCG method is greater than the elapsed time of CG. Note that in this case, although the condensed linear system of equations is the same as the non condensed linear system of equations, all the code to perform the static condensation technique is executed. From degree three, the elapsed time of HCG method is lower than CG.

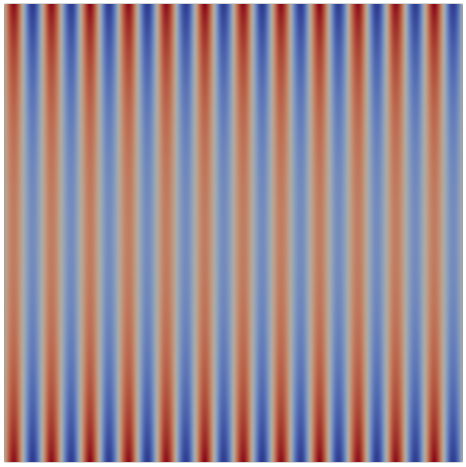


(a) Real Part; Degree $p = 1$

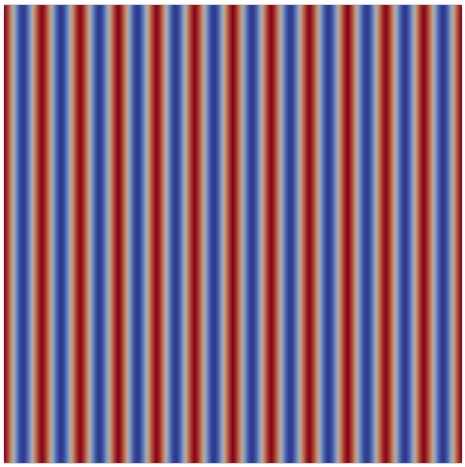
(b) Complex Part; Degree $p = 1$



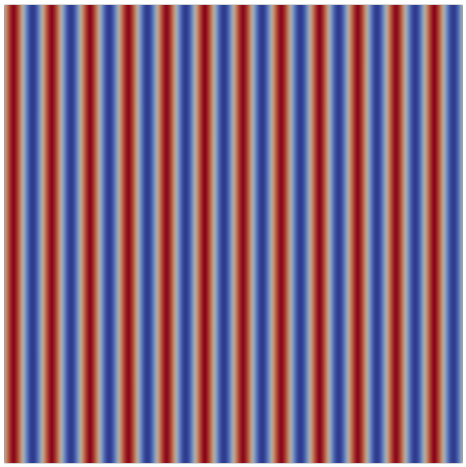
(c) Real Part; Degree $p = 3$



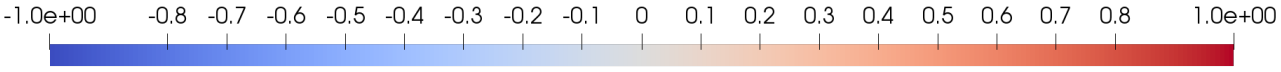
(d) Complex Part; Degree $p = 3$



(e) Real Part; Degree $p = 5$

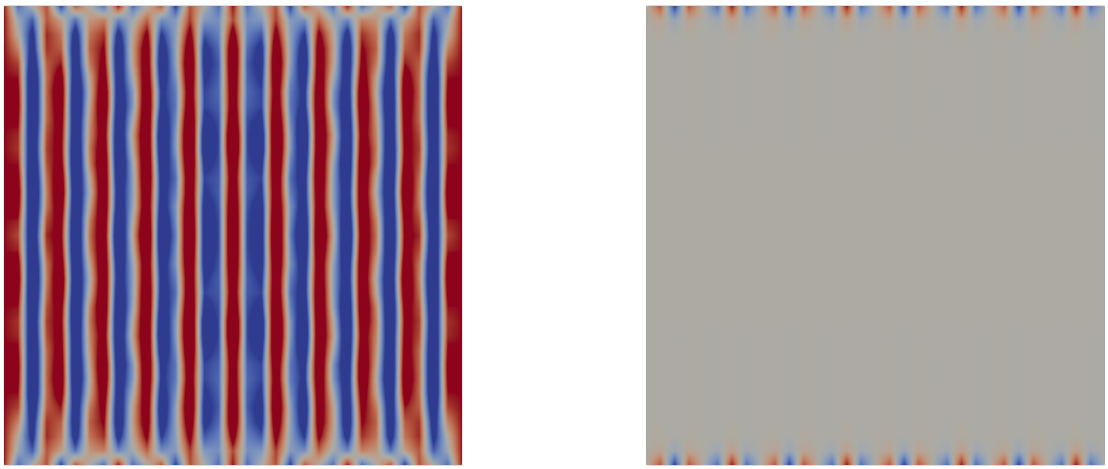


(f) Complex Part; Degree $p = 5$



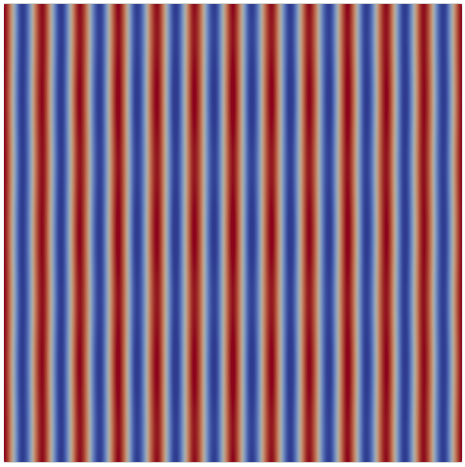
(g) Color Legend

Figure 3: Solution of Helmholtz Equation using CG method with degrees $p = [1, 3, 5]$ and incident wave direction $\mathbf{v} = [1, 0]$

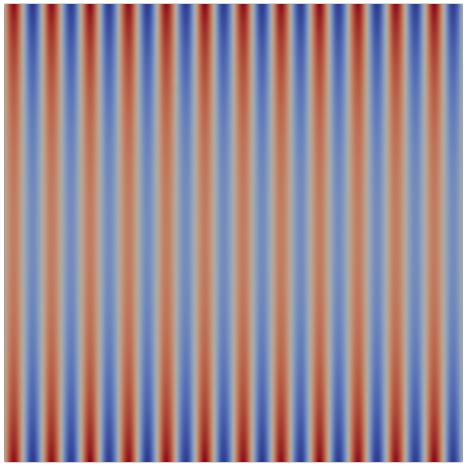


(a) Real Part; Degree $p = 1$

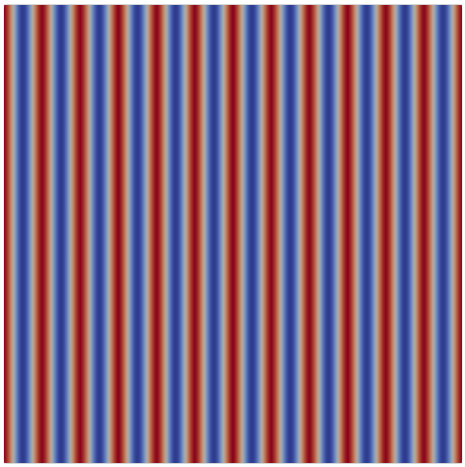
(b) Complex Part; Degree $p = 1$



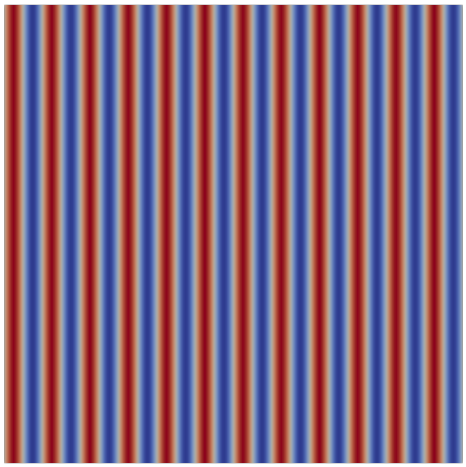
(c) Real Part; Degree $p = 3$



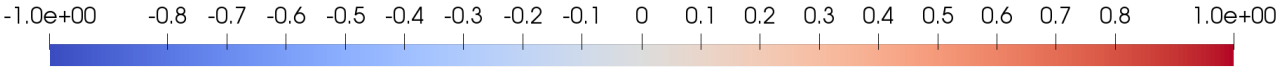
(d) Complex Part; Degree $p = 3$



(e) Real Part; Degree $p = 5$

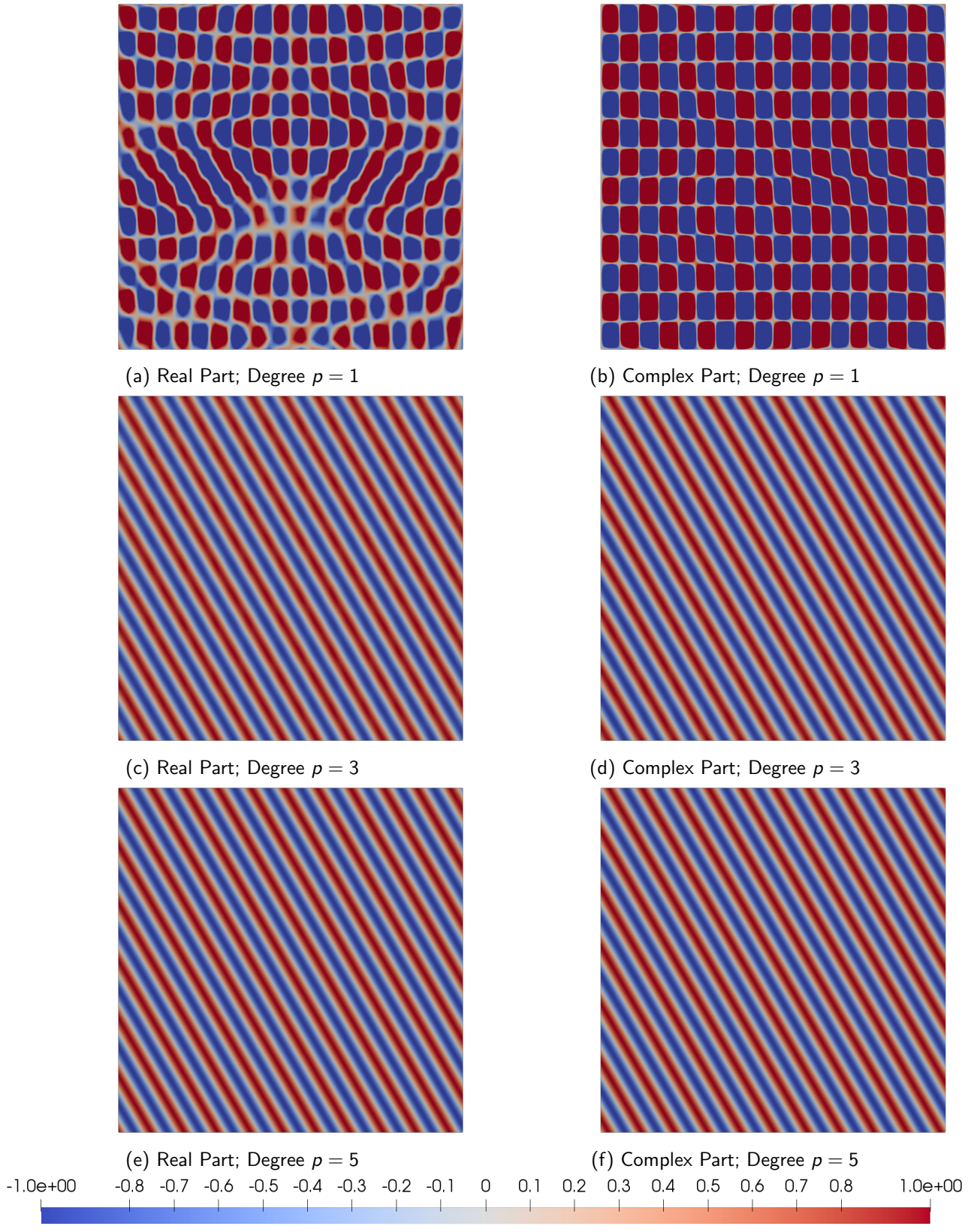


(f) Complex Part; Degree $p = 5$



(g) Color Legend

Figure 4: Solution of Helmholtz Equation using HCG method with degrees $p = [1, 3, 5]$ and incident wave direction $\mathbf{v} = [1, 0]$



(a) Real Part; Degree $p = 1$

(b) Complex Part; Degree $p = 1$

(c) Real Part; Degree $p = 3$

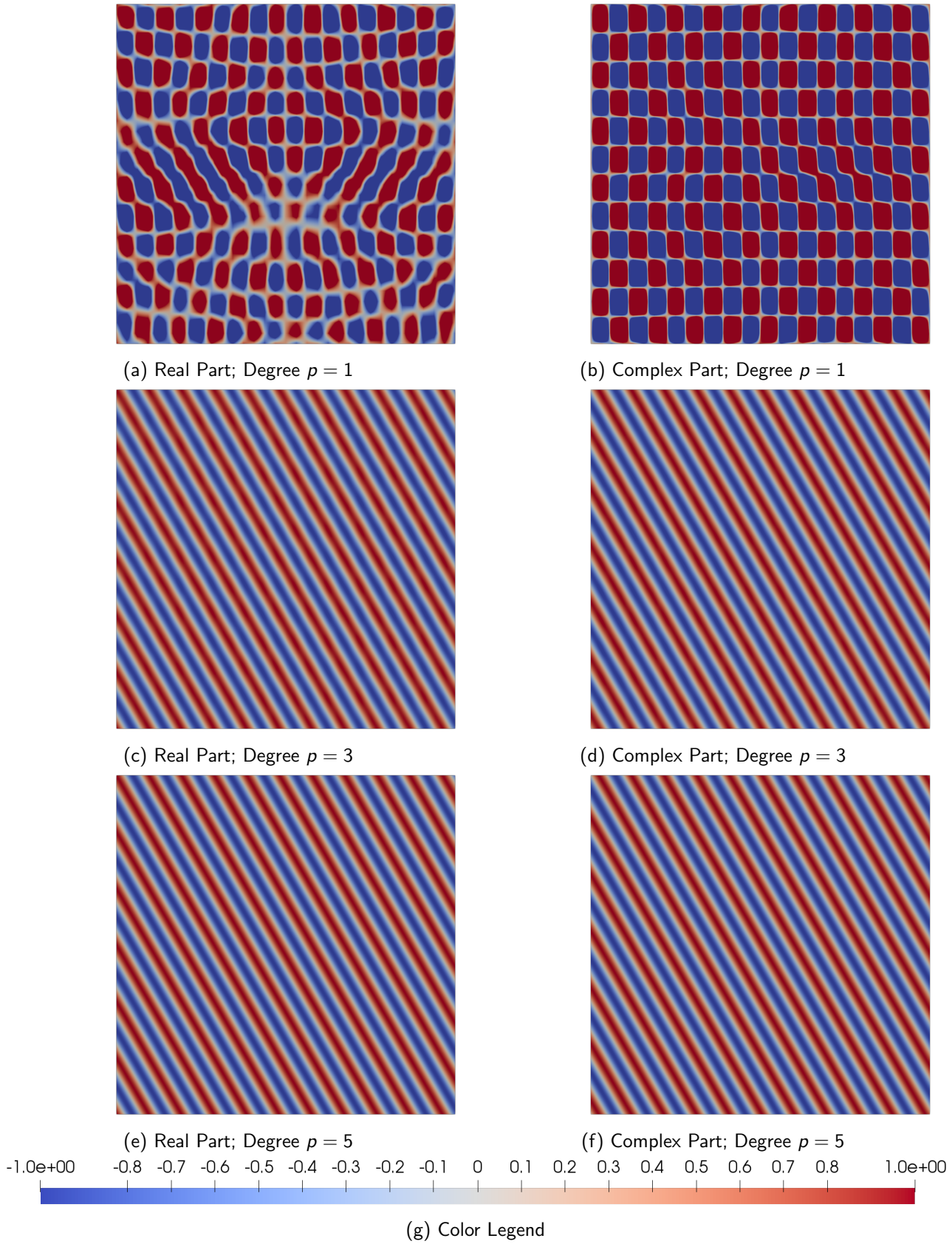
(d) Complex Part; Degree $p = 3$

(e) Real Part; Degree $p = 5$

(f) Complex Part; Degree $p = 5$

(g) Color Legend

Figure 5: Solution of Helmholtz Equation using HCG method with degrees $p = [1, 3, 5]$ and incident wave direction $\mathbf{v} = [0.866, 0.5]$



(a) Real Part; Degree $p = 1$

(b) Complex Part; Degree $p = 1$

(c) Real Part; Degree $p = 3$

(d) Complex Part; Degree $p = 3$

(e) Real Part; Degree $p = 5$

(f) Complex Part; Degree $p = 5$

(g) Color Legend

Figure 6: Solution of Helmholtz Equation using CG method with degrees $p = [1, 3, 5]$ and incident wave direction $\mathbf{v} = [0.866, 0.5]$

4.2 Example 2

For the second example we consider the following problem:

$$\begin{cases} \nabla^2 u + k^2 u = f & \text{in } \Omega \\ \nabla u \cdot \mathbf{n} - i\tau k u = g_R & \text{on } \Gamma_R \end{cases}$$

where $\Omega = [0,1] \times [0,1] \times [0,1]$ and only Robin conditions are considered. The source term is $f(\mathbf{x}) = 0$, the wave number is set to $k = 24\pi$, in order to have 6 wave lengths, and the incident wave directions is $\mathbf{v} = [0.5, -0.5, 0.71]$. Robin condition g_R is determined to verify the analytical solution

$$u(\mathbf{x}) = e^{ik\mathbf{v}\cdot\mathbf{x}}.$$

The mesh considered has 4096 elements and the size of each element is $h = 1/2^4 = 0.0625$.

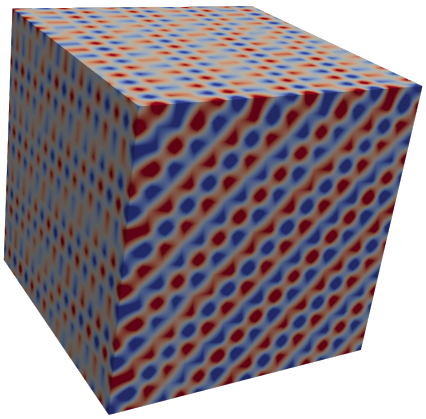
Continuous Galerkin with and without static condensation have been computed using degrees $p = 1, \dots, 4$. Table 3 shows the L^2-norm and the elapsed time of CG and HCG methods varying the degree p from 1 to 4. Figures 8 and 7 show the solution of CG and HCG respectively for a few degrees.

Results show that the error decreases as the degree increments. To be able to find a solution with small error when $k = 24\pi$ we need at least degree 4. The results also show that as the degree increases, the elapsed time of CG method is greater than HCG method. This is because HCG method solves an smaller linear system of equations and the memory usage is lower than in the case of CG method. We have shown the same result in the first example but the difference is more significant when the domain is three-dimensional.

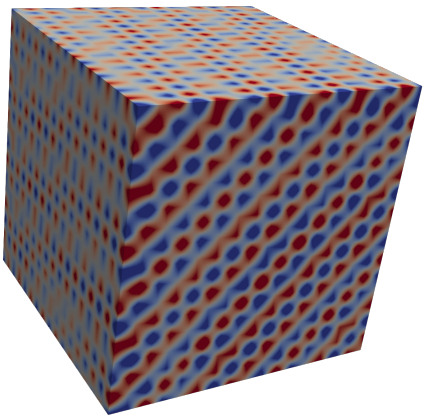
We have exactly the same solution using CG and HCG. When the degree is $p = 1$ the solution is not well captured as we can see at figures 7b and 7a. When the degree is $p = 2$, we can observe at figures 7d and 7c that there are some perturbations at the bottom of the right hand side of the cube. When degree is $p = 4$ the solution is well captured and the error is small enough.

			CG		HCG	
Degree	Nodes	Elements	Error	Time (s)	Error	Time (S)
1	4913	4096	1.143E+00	2.108E+01	1.143E+00	2.055E+01
2	35937	4096	1.075E+00	1.001E+03	1.075E+00	9.745E+02
3	117649	4096	1.286E-01	1.239E+04	1.286E-01	8.18E+03
4	274625	4096	8.306E-03	1.223E+05	8.306E-03	5.881E+04

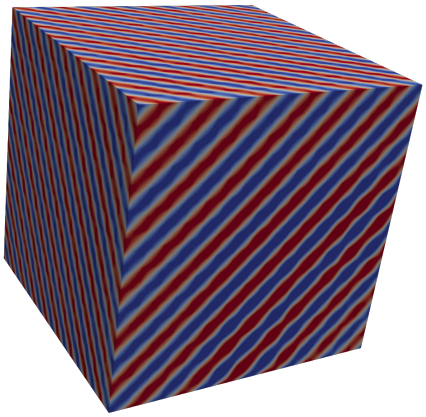
Table 3: CG and HCG solution with $p = 1, \dots, 4$, $h = 1/2^4$



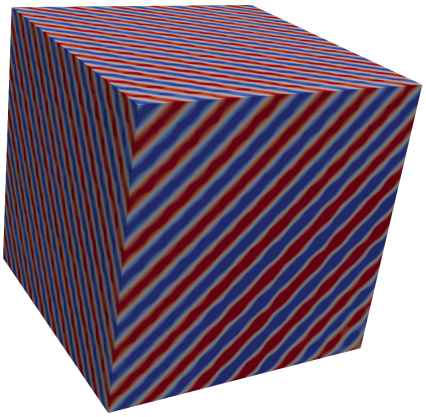
(a) Real Part; Degree $p = 1$



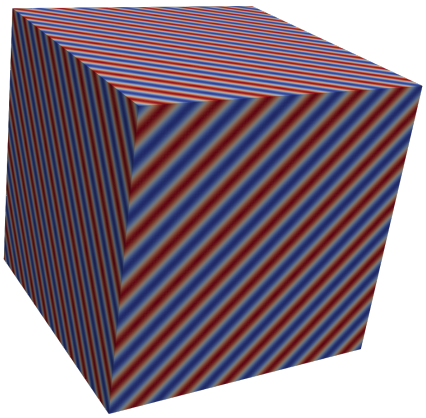
(b) Complex Part; Degree $p = 1$



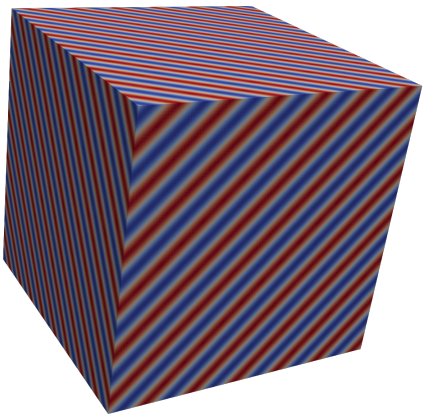
(c) Real Part; Degree $p = 2$



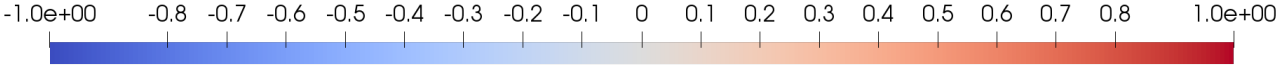
(d) Complex Part; Degree $p = 2$



(e) Real Part; Degree $p = 4$

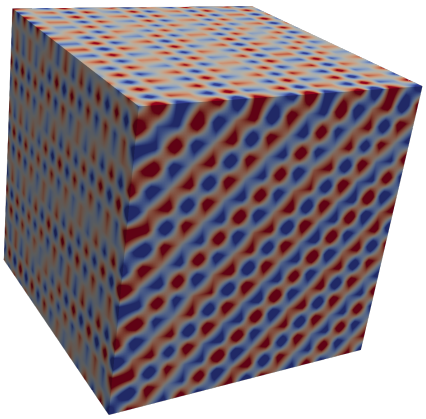


(f) Complex Part; Degree $p = 4$

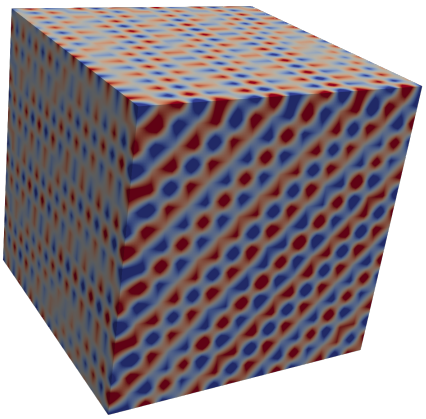


(g) Color Legend

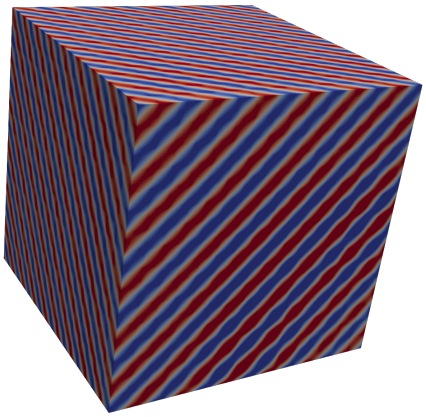
Figure 7: Solution of Helmholtz Equation using HCG method with degrees $p = [1, 2, 4]$ and incident wave direction $\mathbf{v} = [0.5, -0.5, 0.71]$



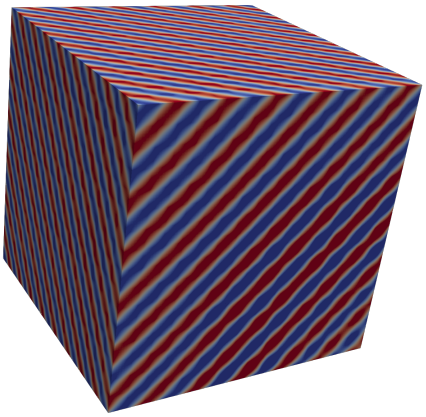
(a) Real Part; Degree $p = 1$



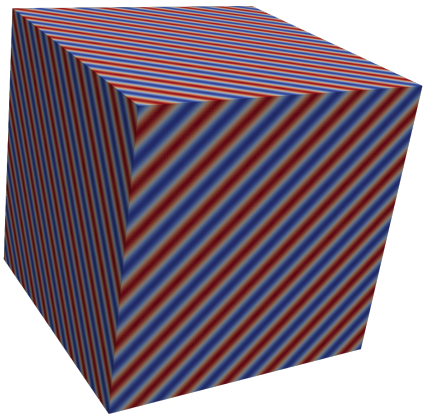
(b) Complex Part; Degree $p = 1$



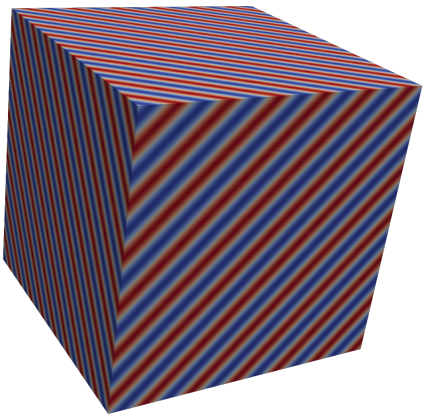
(c) Real Part; Degree $p = 2$



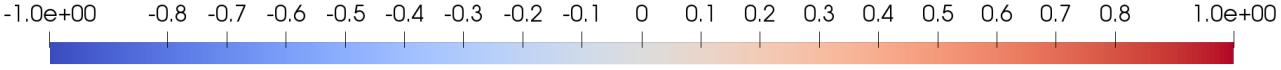
(d) Complex Part; Degree $p = 2$



(e) Real Part; Degree $p = 4$



(f) Complex Part; Degree $p = 4$



(g) Color Legend

Figure 8: Solution of Helmholtz Equation using CG method with degrees $p = [1, 2, 4]$ and incident wave direction $\mathbf{v} = [0.5, -0.5, 0.71]$

4.3 Convergence Analysis

The asymptotic behaviour of the error, for small h is:

$$E_u \sim Ch^{p+1}$$

The convergence analysis has been performed for the solution of the Helmholtz equation for the continuous galerkin method and hybridizable continuous galerkin method using Robin and Dirichlet boundary conditions. Both domains, $\mathbb{R}^2$ and $\mathbb{R}^3$,will be considered.

We consider a series of quadrilateral meshes such that the length of each element is $h^S = 1/2^S$ for $S = 1, \dots, 5$ and polynomial degrees ranging for $p = 1, \dots, 5$. The wave number is set to $k = 2\pi$ and the incident wave direction is set to $\mathbf{v} = [1, 0, 0]$ for $\mathbb{R}^3$ and $\mathbf{v} = [1, 0]$ for $\mathbb{R}^2$. The boundary conditions are determied in such a way that the analytical solution is $u(\mathbf{x}) = e^{ik\mathbf{v}\cdot\mathbf{x}}$. Then, for each mesh we compute the numerical solution using continuous and hybridizable continuous Galerkin method.

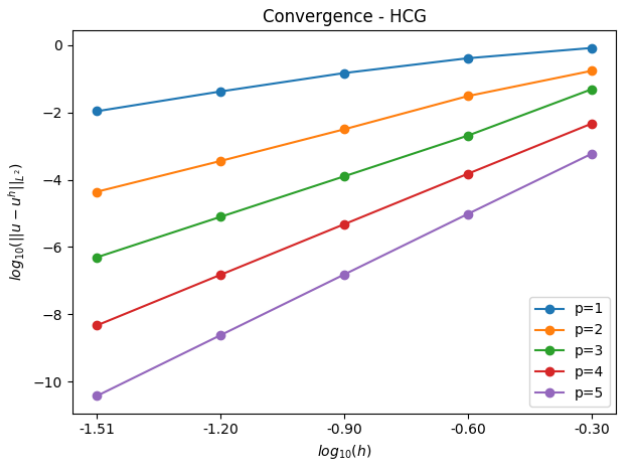
Assimptotically, the numerical error behaves as $E_u \sim Ch^{p+1}$, for h small. Thus we expect that,

$$log(e) = log(C) + (p+1)log(h),$$

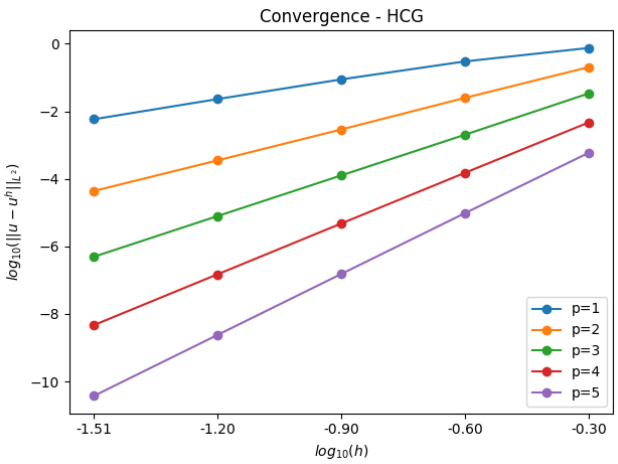
which is an affine function with slope $p + 1$.

Since the L_2-Norm is the same for CG and HCG methods, figure 9 shows the graphic results of the error for the HCG method using Dirichlet and Robin boundary conditions in 2D and 3D. At table 4 we see for each S and integration degree p the error and the order of convergence for each method using Dirichlet boundary conditions and Robin boundary conditions in 2D. Table 5 presents the same information when the domain is 3D.

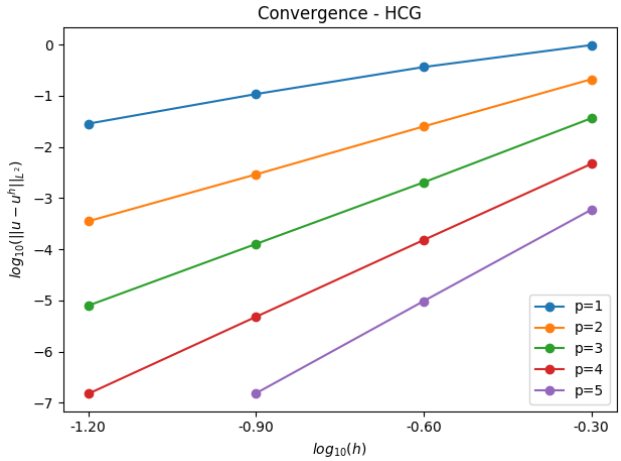
We observe that as the size of the element decreases, s increses, and as the degree increases the error decreases. In general the slope observed is approximately $p + 1$, when h is small enough. There isn't significant differences in the order when Dirichlet or Robin boundary conditions are used or when we change the dimension of the domain. We can observe that, when h is not small enough, the convergence order is lower than $p+1$. We can conclude that the numerical convergence analysis verify that the implementation performs well.



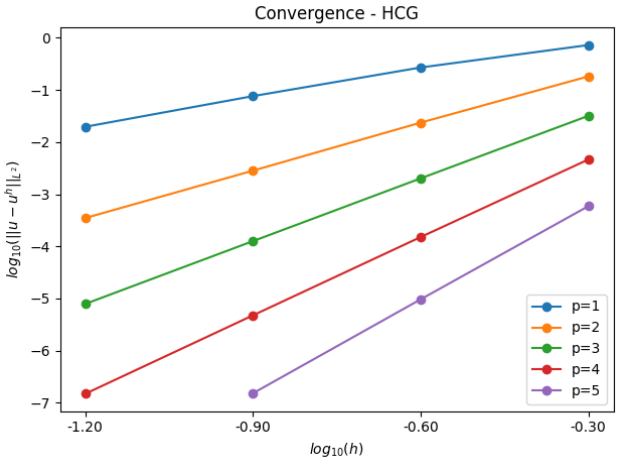
(a) Dirichlet conditions 2D



(b) Robin conditions 2D



(c) Dirichlet conditions 3D



(d) Robin conditions 3D

Figure 9: Error of solving Helmholtz equation using CG method with interpolation degree p, $1 \leq p \leq 5$, and element size $h^s = 1/2^s$ for $1 \leq s \leq 5$

		Dirichlet				Robin			
		CG		HCG		CG		HCG	
S	Degree	Error	Order	Error	Order	Error	Order	Error	Order
1	1	8.280E-01	-	8.280E-01	-	7.511E-01	-	7.511E-01	-
1	2	4.103E-01	1.01	4.103E-01	1.01	3.000E-01	1.32	3.000E-01	1.32
1	3	1.489E-01	1.46	1.489E-01	1.46	8.806E-02	1.77	8.806E-02	1.77
1	4	4.213E-02	1.82	4.213E-02	1.82	2.301E-02	1.94	2.301E-02	1.94
1	5	1.089E-02	1.95	1.089E-02	1.95	5.817E-03	1.98	5.817E-03	1.98
2	1	1.732E-01	-	1.732E-01	-	2.028E-01	-	2.028E-01	-
2	2	3.079E-02	2.49	3.079E-02	2.49	2.501E-02	3.02	2.501E-02	3.02
2	3	3.163E-03	3.28	3.163E-03	3.28	2.895E-03	3.11	2.895E-03	3.11
2	4	3.613E-04	3.13	3.613E-04	3.13	3.519E-04	3.04	3.519E-04	3.04
2	5	4.395E-05	3.04	4.395E-05	3.04	4.365E-05	3.01	4.365E-05	3.01
3	1	4.900E-02	-	4.900E-02	-	3.379E-02	-	3.379E-02	-
3	2	2.056E-03	4.57	2.056E-03	4.57	2.026E-03	4.06	2.026E-03	4.06
3	3	1.264E-04	4.02	1.264E-04	4.02	1.263E-04	4.00	1.263E-04	4.00
3	4	7.895E-06	4.00	7.895E-06	4.00	7.894E-06	4.00	7.894E-06	4.00
3	5	4.935E-07	4.00	4.935E-07	4.00	4.935E-07	4.00	4.935E-07	4.00
4	1	4.669E-03	-	4.669E-03	-	4.678E-03	-	4.678E-03	-
4	2	1.509E-04	4.95	1.509E-04	4.95	1.509E-04	4.95	1.509E-04	4.95
4	3	4.762E-06	4.99	4.762E-06	4.99	4.762E-06	4.99	4.762E-06	4.99
4	4	1.492E-07	5.00	1.492E-07	5.00	1.492E-07	5.00	1.492E-07	5.00
4	5	4.665E-09	5.00	4.665E-09	5.00	4.665E-09	5.00	4.665E-09	5.00
5	1	5.917E-04	-	5.917E-04	-	5.906E-04	-	5.906E-04	-
5	2	9.644E-06	5.94	9.644E-06	5.94	9.644E-06	5.94	9.644E-06	5.94
5	3	1.524E-07	5.98	1.524E-07	5.98	1.524E-07	5.98	1.524E-07	5.98
5	4	2.388E-09	6.00	2.388E-09	6.00	2.388E-09	6.00	2.388E-09	6.00
5	5	3.734E-11	6.00	3.734E-11	6.00	3.734E-11	6.00	3.734E-11	6.00

Table 4: Error of solving Helmholtz equation using CG and HCG methods with Dirichlet and Robin boundary conditions with two-dimensionl domain

		Dirichlet				Robin			
		CG		HCG		CG		HCG	
S	Degree	Error	Order	Error	Order	Error	Order	Error	Order
1	1	9.737E-01	-	9.737E-01	-	7.260E-01	-	7.260E-01	-
1	2	3.623E-01	1.43	3.623E-01	1.43	2.697E-01	1.43	2.697E-01	1.43
1	3	1.074E-01	1.75	1.074E-01	1.75	7.621E-02	1.82	7.621E-02	1.82
1	4	2.833E-02	1.92	2.833E-02	1.92	1.967E-02	1.95	1.967E-02	1.95
2	1	2.091E-01	-	2.091E-01	-	1.819E-01	-	1.819E-01	-
2	2	2.499E-02	3.06	2.499E-02	3.06	2.362E-02	2.95	2.362E-02	2.95
2	3	2.889E-03	3.11	2.889E-03	3.11	2.836E-03	3.06	2.836E-03	3.06
2	4	3.516E-04	3.04	3.516E-04	3.04	3.499E-04	3.02	3.499E-04	3.02
3	1	3.614E-02	-	3.614E-02	-	3.185E-02	-	3.185E-02	-
3	2	2.021E-03	4.16	2.021E-03	4.16	2.014E-03	3.98	2.014E-03	3.98
3	3	1.263E-04	4.00	1.263E-04	4.00	1.262E-04	4.00	1.262E-04	4.00
3	4	7.894E-06	4.00	7.894E-06	4.00	7.894E-06	4.00	7.894E-06	4.00
4	1	4.634E-03	-	4.634E-03	-	4.628E-03	-	4.628E-03	-
4	2	1.507E-04	4.94	1.507E-04	4.94	1.508E-04	4.94	1.508E-04	4.94
4	3	4.761E-06	4.98	4.761E-06	4.98	4.761E-06	4.99	4.761E-06	4.99
4	4	1.492E-07	5.00	1.492E-07	5.00	1.492E-07	5.00	1.492E-07	5.00
5	1	5.892E-04	-	5.892E-04	-	5.890E-04	-	5.890E-04	-
5	2	9.642E-06	5.93	9.642E-06	5.93	9.642E-06	5.93	9.642E-06	5.93
5	3	1.524E-07	5.98	1.524E-07	5.98	1.524E-07	5.98	1.524E-07	5.98

Table 5: Error of solving Helmholtz equation using CG and HCG methods with Dirichlet and Robin boundary conditions with three-dimensionl domain

4.4 Example 3

In this section we present the scattering of a plane wave by a rigid body citeihlenburg2006finite. We assume that the solution of the problem is composed of a prescribed incident wave, u_I, plus a scattered wave u_R,

$$u_T = u_I + u_R,$$

where the incident wave is of the form $u_I = e^{ik\mathbf{v}\cdot\mathbf{x}}$. We assume that there are no sources in the domain, $f = 0$ and a perfect reflection is obtained on the rigid body, $\nabla u_T \cdot \mathbf{n} = 0$.

Since u_T is a wave, it has to satisfy the Helmholtz equation

$$\begin{cases} \nabla^2 u_T + k^2 u_T = 0 & \text{in } \Omega \\ \nabla u_T \cdot \mathbf{n} = 0 & \text{on } \Gamma_N \\ \nabla u_T \cdot \mathbf{n} - iku_T = 0 & \text{on } \Gamma_R. \end{cases}$$

Moreover, the incident wave u_I satisfies the homogeneous Helmholtz equation.

Thus, the complex Helmholt equation is to find $u_R : \Omega \to \mathbb{C}$ such that the reflected wave u_R satisfies the following problem

$$\begin{cases} \nabla^2 u_R + k^2 u_R = 0 & \text{in } \Omega \\ \nabla u_R \cdot \mathbf{n} = iku_I \mathbf{v} \cdot \mathbf{n} & \text{on } \Gamma_N \\ \nabla u_R \cdot \mathbf{n} - iku_R = 0 & \text{on } \Gamma_R \end{cases}$$

Notice that, since Neumann condition for the total wave is $\nabla u_T \cdot \mathbf{n} = 0$, in terms of the incident wave we have that $\nabla u_I \cdot \mathbf{n} + \nabla u_R \cdot \mathbf{n} = 0$. Then, Neumann condition for the reflected wave should verify that $\nabla u_R \cdot \mathbf{n} = iku_I \mathbf{v} \cdot \mathbf{n}$. The same strategy used to find Neumann boundary conditions could give us the Robin boundary conditions. But we set Robin boundary conditions equal to zero to represent that the boundary is far enough from the object.

We consider a square domain $[-1, 2]x[-1, 1]$ with Robin boundary conditions at the edges an Neumann boundary conditions at the boundary of the object. See figure 10. The mesh has 24430 elements of size $1/2^6$ and we discretize the domain using elements of polynomial degree 5. Two incident wave directions are tested: $v = [0.855, 0.5]$ and $v = [1.0, 0.0]$. The wave number is set to be $k = 48\pi$.

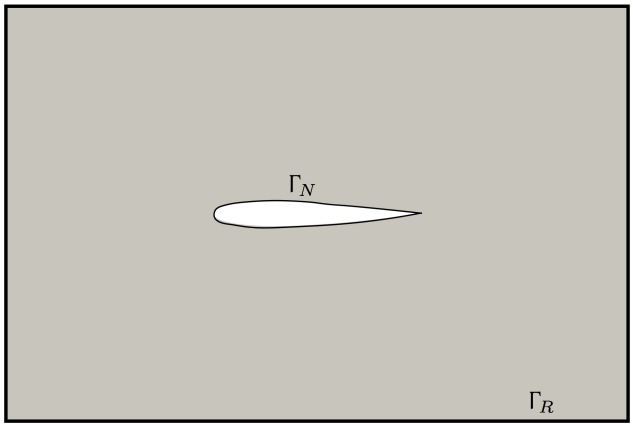
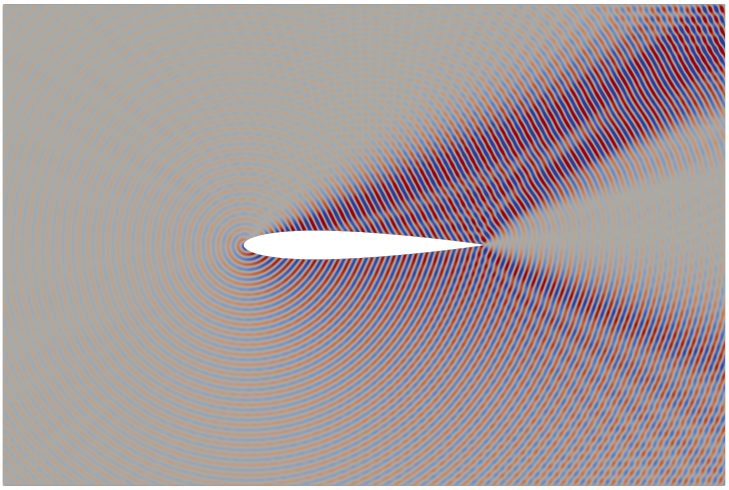


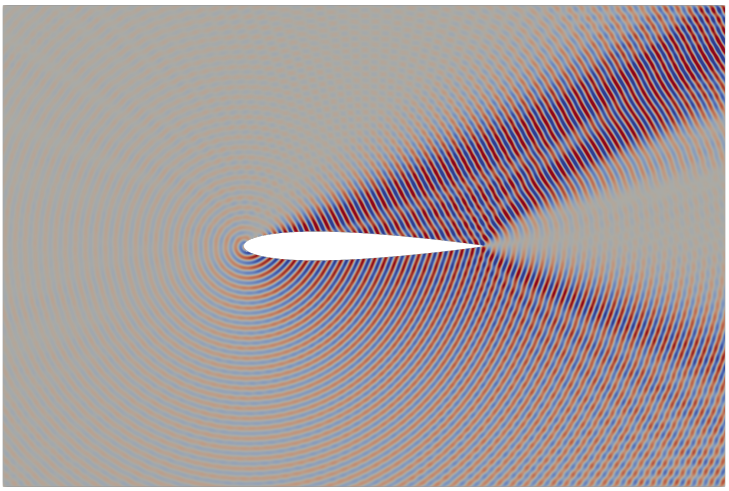
Figure 10: Example 3 mesh

Figures 11, 12 show the reflected wave u_R as the result of solving (4.4) using hybridizable continuous Galerkin Method. Since we are using high-order polynomials, the solution is smooth enough around the airfoil profile. The total number of unknowns is 612675. When static condensation is used, the number of unknowns is reduced to 221795.

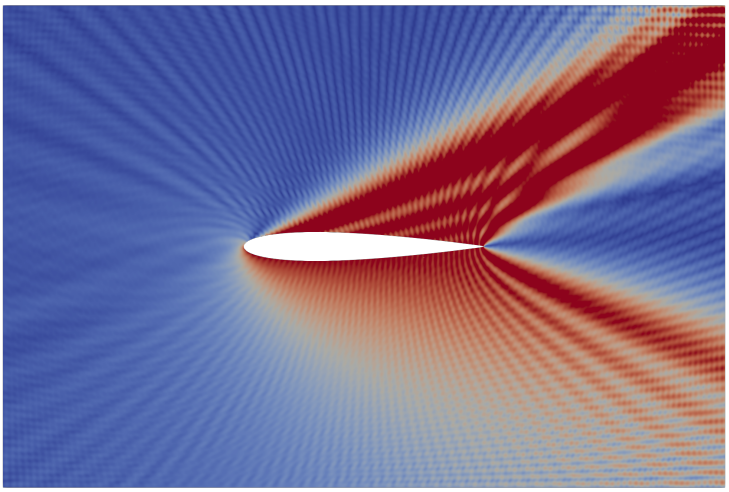
In that case, the analytical solution is unknown and the L^2 norm can not be computed. The elapsed time to solve that problem using the implemented HCG method is 2262.68 s.



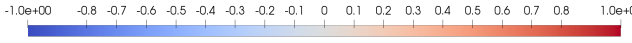
(a) Real Part



(b) Complex Part



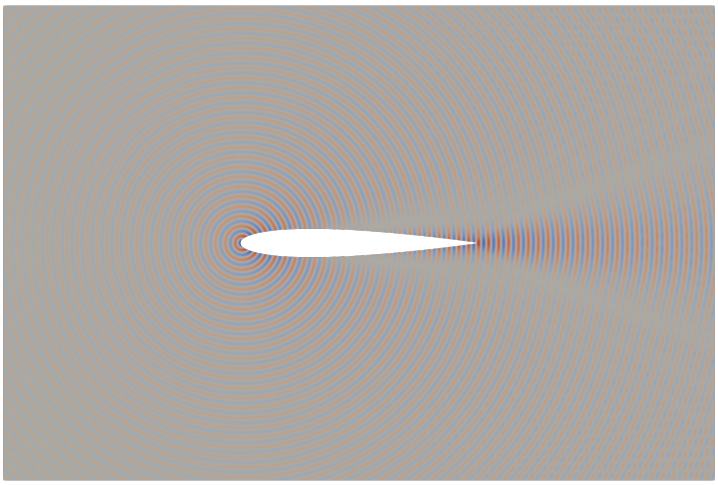
(c) Modul



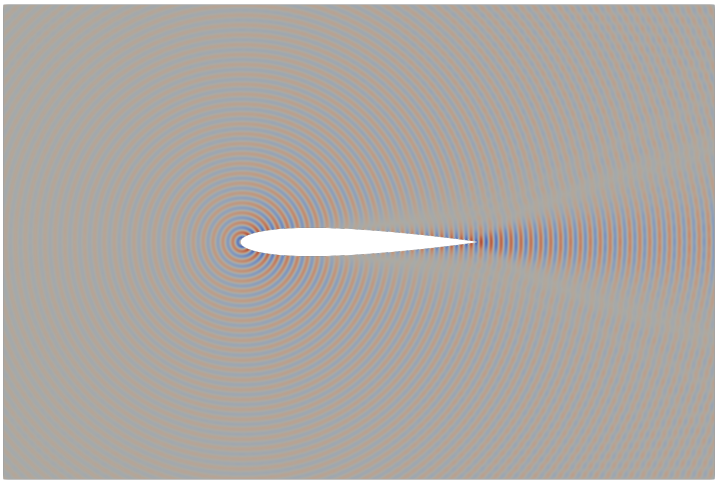
(d) Color Legend for the real and the complex part

(e) Color Legend for the modul

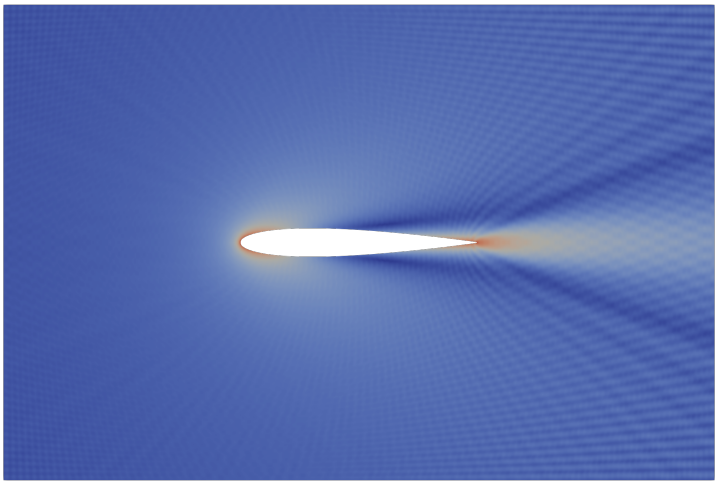
Figure 11: Solution of Helmholtz Equation using HCG method with degree $p = 5$, $k = 48\pi$ and incident wave direction $\mathbf{v} = [0.866, 0.5]$



(a) Real Part



(b) Complex Part



(c) Modul

(d) Color Legend for the real and the complex part

(e) Color Legend for the modul

Figure 12: Solution of Helmholtz Equation using HCG method with degree $p = 5$, $k = 48\pi$ and incident wave direction $\mathbf{v} = [1.0, 0.0]$

5. Conclusions and Future work

Two finite element methods have been implemented to solve wave problems that satisfy the Helmholtz equation. Continuous Galerkin method has been improved in terms of efficieny using the static condensation technique. The static condensation technique reduces the number of unknowns of the linear system. Due to that fact the hybridizable continuous Galerking method reduces the amount of memory usage and the elapsed execution time respect to the classical continuous Galerking Method.

Continuous Galerkin formulation with high-order polynomials has exponential convergence of the numerical error. The convergence analysis validates our implementation because it verifies numerically that the order of convergence of CG and HCG methods is $p + 1$ for small element size h. Thus, the numerical errors obtained in the tests are in accordance with the theoretical error bounds.

Continuous Galerkin formulation can be used with high-order meshes. For instance, Example 3 shows the result of using a high-order curved mesh which gives us a smooth solution.

Continuous Galerkin formulation with high-order polynomials involves larger linear systems than low order formulations, where the matrix is less sparse. Due to that fact, high order formulations increment the elapsed time and the memory usage respect to low order formulations.

There are at least two ways to continue this master thesis. Firstly the execution time of the HCG method can be improved because the small linear system of equations that has to be solved for each element can be computed in parallel. Once the large system of equations for the unknowns at the boundary is solved there are no dependencies between the small linear systems. Another improvement could be use iterative methods to solve the linear system of equations and introduce a preconditioner in order to improve the convergence of the method. The usage of iterative methods and preconditioners will also improve the memory usage and elapsed time.

References

[1] Continuum Analytics. Anaconda software distribution. computer software. vers. 2-2.4.0.

[2] James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, and Joseph W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications*, 20(3):720–755, 1999.

[3] Arnaud Deraemaeker, Ivo Babuška, and Philippe Bouillard. Dispersion and pollution of the fem solution for the helmholtz equation in one, two and three dimensions. *International journal for numerical methods in engineering*, 46(4):471–499, 1999.

[4] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[5] Frank Ihlenburg. *Finite element analysis of acoustic scattering*, volume 132. Springer Science & Business Media, 2006.

[6] Intel. Developer reference for intel math kernel library c.

[7] Intel. Lapack routines.

[8] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[9] Robert M Kirby, Spencer J Sherwin, and Bernardo Cockburn. To cg or to hdg: a comparative study. *Journal of Scientific Computing*, 51(1):183–212, 2012.

[10] G. van Rossum. Python software foundation. python language reference, version 2.7.

[11] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[12] Olgierd Cecil Zienkiewicz, Kenneth Morgan, and Kenneth Morgan. *Finite elements and approximation*. Courier Corporation, 2006.